

A Generic Data Model for Schema-driven Design in Information Retrieval Applications

Hany Azzam, and Thomas Roelleke

Queen Mary, University of London, UK. E1 4NS.
{hany,thor}@eeecs.qmul.ac.uk

Abstract. Database technology offers design methodologies to rapidly develop and deploy applications that are easy to understand, document and teach. It can be argued that information retrieval (IR) lacks equivalent methodologies. This poster discusses a generic data model, the Probabilistic Object-Oriented Content Model, that facilitates solving complex IR tasks. The model guides how data and queries are represented and how retrieval strategies are built and customised. Application/task-specific schemas can also be derived from the generic model. This eases the process of tailoring search to a specific task by offering a layered architecture and well-defined schema mappings. Different types of knowledge (facts and content) from varying data sources can also be consolidated into the proposed modelling framework. Ultimately, the data model paves the way for discussing IR-tailored design methodologies.

1 Introduction & Motivation

Nowadays, large-scale knowledge bases can be automatically generated from high-quality knowledge sources such as Wikipedia and other semantically explicit data repositories such as ontologies and taxonomies that explain entities (e.g. mark-up of persons, movies, locations and organisations) and record relationships (e.g. bornIn, actedIn and isCEOof). Such knowledge bases are leveraged by information retrieval (IR) application developers to develop more semantically-aware retrieval systems as opposed to systems that utilise text only. However, the developed systems are usually tailored to a particular data format and/or application. This is problematic since developing new applications or incorporating new data formats usually requires “reimplementing APIs, introducing new APIs, introducing new query languages, and even introducing new indexing and storage structures” [3].

The question, thus, becomes how diverse applications and data formats can be supported by a single unifying framework. Additionally, how techniques developed for a particular data format such as text can be easily transferred/extended to other data formats. This poster attempts to answer these two questions. We propose a generic data model that facilitates the development process of IR applications. The data model represents facts (e.g. objects and their relationships) and content knowledge (e.g. text in documents) in one congruent data model. The model can also be used to transfer text retrieval models such as TF-IDF, language modelling (LM) and BM25 to more knowledge-oriented retrieval models. Finally, the model can facilitate the expression of more complex and semantically expressive representations of information needs.

2 The Data Model

The proposed data model, the Probabilistic Object-Oriented Content Model (POOCM), combines 1) probability theory, 2) object-oriented modelling and 3) content-oriented modelling into one framework. The POOCM consists of term, classification, relationship and attribute propositions. Additionally, in order to perform content-oriented reasoning (traditional IR), each predicate has a context (context refers to documents, sections, databases and any other object with content).

A distinctive characteristic of this data model is that unlike standard artificial intelligence and database approaches content is modelled via a concept separate from the existing object-oriented concepts (classifications, relationships and attributes). This keeps the design tidy and captures the distinctive characteristics of each of the concepts, i.e. it enables the construction of evidence spaces based on each of the modelling concepts. The following representation of the movie “Apocalypse Now” illustrates the nature of the POOCM. The example shows two possible syntactic formulations: one based on predicate logic (e.g. Datalog), and the other similar to terminological logics [4].

```
# Term 'vietnam' in movie_329171
0.5 vietnam(movie_329171);           # movie_329171[0.5 vietnam]
# Classification 'marlon_brando is an actor' in imdb
0.7 actor(marlon_brando, imdb);     # imdb[0.7 actor(marlon_brando)]
# Classification 'walter_kurtz is a colonel' in movie_320971
colonel( walter_kurtz , movie_3209171); # movie_329171[colonel( walter_kurtz )]
# Relationship 'marlon_brando playsRoleOf walter_kurtz ' in movie_329171
playsRoleOf(marlon_brando, walter_kurtz , movie_329171);
# Attribute 'movie_329171 has release date 1979' in imdb
hasReleaseDate(movie_329171, 1979, imdb); # movie_329171.hasReleaseDate(1979)
```

From an entity-relationship modelling point of view, the POOCM generally represents relationships between objects, relationships between classes and relationships between objects and classes. However, unlike the entity-relationship model, the POOCM incorporates content-oriented modelling techniques and concepts of probability theory which lead to a data model that is tailored to solving IR applications/tasks. The probabilities can be based on frequencies such as those commonly used in IR models.

The data model allows the handling of the physical data structures to remain transparent for (decoupled from) the rest of the system design, thus achieving what the DB field calls ‘data independence’ [2, 3]. Furthermore, it enables the development of retrieval models that leverage the underlying data while remaining independent of the physical data representation. This is a desirable feature for designing complex retrieval systems as it ensures the independence of the developed retrieval models and query languages from the actual document representation [1].

Another benefit is that the object-oriented and content-oriented concepts of the POOCM provide the ability to instantiate retrieval models comprised of term, classification, relationship and attribute propositions. This leads to knowledge-oriented retrieval models that exploit a particular type of evidence explicated by the propositions. On the information need side, the data model can enrich query representation which facilitates the expression of more complex and expressive representations of information needs.

3 Modelling Layers

Application-independent and application-specific schemas can be instantiated from the generic POOCM. A simplified structure of the model distinguishes between three modelling layers: basic, structural and semantic layer. *Layer 0* (the basic layer) is *application-independent*, and the upper layers are more application-specific.

Generally speaking, overly specific schemas (e.g. fully flagged and normalised relational schemas as proposed by traditional DB design) and overly general schemas (triplet storages) are two extremes for IR. The POOCM does not argue that one approach is better than another, but demonstrates how application-specific schema layers can be derived from more general/basic ones.

Layer 1 is the *element-based* layer. It contains rules that can derive structural predicates from the L0, and the structural object Ids are made explicit. These rules can “lift” the basic classifications and attributes into structural classifications and attributes.

Layer 2 is the *entity-based* layer. It contains rules that derive semantic classification and relationships. For example, the rules extract objects by combining structural information about element types and their attributes. Such modelling of entities is prevalent in Entity-Relationship-graphs, such as RDF, where URIs are used to denote objects.

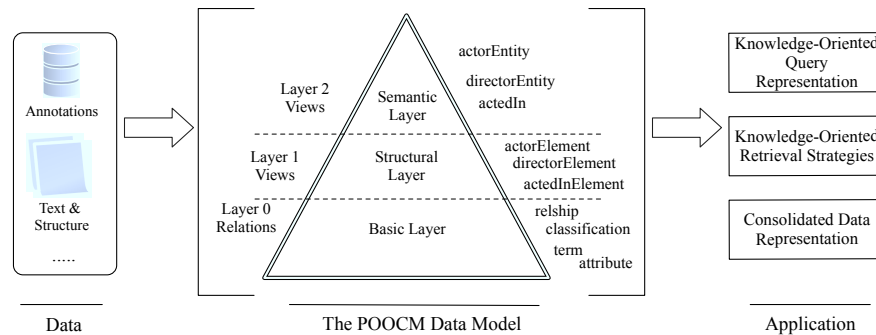


Fig. 1. POOCM Layers: Basic, Structural and Semantic

Fig. 1 highlights the main schema layers. These layers form an abstraction hierarchy that helps to achieve data independence. Any data format (e.g. XML, RDF, text) can be represented using the application-independent and application-specific schemas.

Another advantage of this layered approach is that explicitly stating how the basic and semantic layers are related can impact the modelling of probability estimations and aggregations. The predicates in the basic layer can, for example, be used to construct an evidence space for term-based retrieval models (e.g. LM) and for basic semantic models (e.g. attribute-based LM). In the structural and semantic layers, however, more complex and tailored (application-specific) models can be constructed while maintaining the advocated reusability and ability to be customised.

4 Probability Estimation

Probabilities used to develop IR models are an inherent component of the schema. The POOCM can guide which and how probabilities are estimated. For example, L1 consists of relations tailored to modelling the structure of data. This includes relations for context-based segmentations, e.g. $\text{term_doc}(\text{Term}, \text{Doc})$ to index documents, and $\text{term_sec}(\text{Term}, \text{Sec})$ to index sections. Probabilistic relations can be derived for each of the L1 relations. L2 comprises of relations reflecting the semantics of the data (semantic lifting of L1 leads to L2 relations). For instance, “actedIn(Actor,Movie,Context)” is L2. Note that L2 relation names bear a meaning, L1 relation names indicate the type of the context, and L0 relation names reflect classifications and relationships.

For each relation in each layer there are probabilistic relations for the sets of attributes. The probabilities can be value- or tuple-based. As such, the concepts of IR naturally apply to the semantic and generic schema. Concepts such as the tuple frequency of a class or the IDF of a class name make immediate sense. The following illustrates some of these probabilistic relations.

- $P_{VF}(t|i)$: Value-Frequency-based probability of term t derived from an index i such as “term(Term, Doc)” where the occurrence in documents (values) is the evidence.
- $P_{TF}(t|i)$: Tuple-Frequency-based probability of term t derived from an index i where the occurrence in locations (tuples) is the evidence.
- $P_{IVF}(t|i)$: IVF-based (IVF: inverse value frequency) probability of term t , e.g. $-\log P_{VF}(t|i) / \max\{-\log P_{VF}(t'|i)\}$. For document retrieval $\text{IDF}=\text{IVF}$, and for actor retrieval $\text{InvActorFreq}=\text{IVF}$.

5 Conclusion

The generic data model (POOCM) advocated in this poster supports the design process when solving different IR tasks. The role of the model can be compared to what terminological logic [4] is for modelling knowledge: a conceptual quasi-standard that offers guidance while eschewing syntactical constraints. This poster aims at initiating a discussion about the role of the “design process” in IR - a process that so far has not been guided by an IR-tailored methodology. The hypothesis is that IR urgently needs such a methodology to respond to the growing need for the management of complex engineering processes and diverse content representations.

References

1. R. Cornacchia and A. de Vries. A parameterised search system. In *ECIR*, 2007.
2. N. Fuhr. Towards data abstraction in networked information retrieval systems. *IP&M*, 35(2):101–119, 1999.
3. D. Hiemstra and V. Mihajlovic. A database approach to information retrieval: The remarkable relationship between language models and region models. *CTIT Technical Report*, 2010.
4. C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *SIGIR*, 1993.